



Katedra za Signale i sisteme
Elektrotehnički fakultet
Univerzitet u Beogradu



13E054PO - Prepoznavanje oblika

Domaći zadatak

Aleksa Janjić 2019/0021

Sadržaj

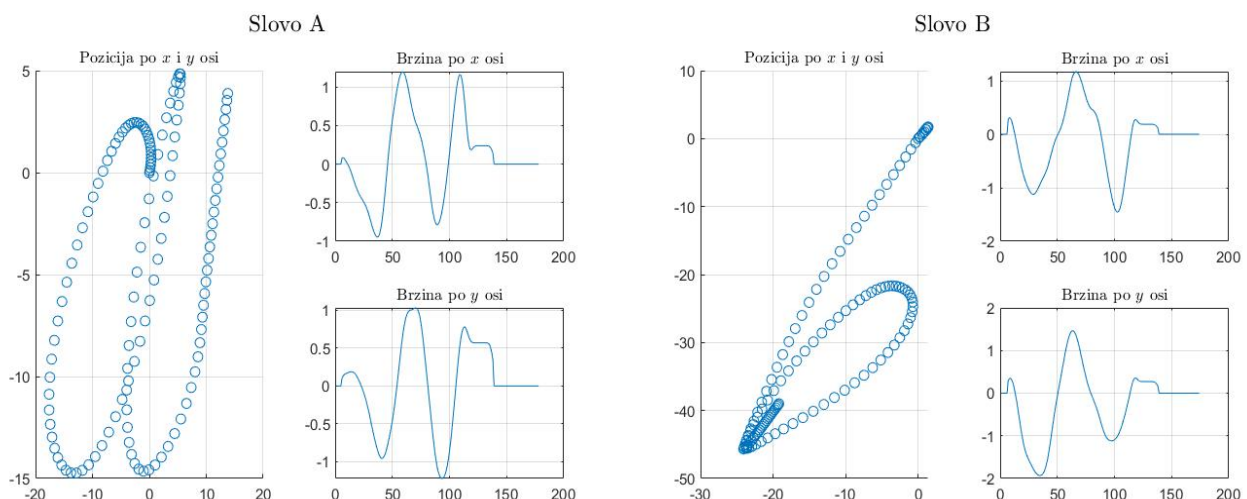
1	Zadatak 1 - Opcija 1	2
2	Zadatak 2	8
3	Zadatak 3 - Opcija 1	15
4	Zadatak 4	20

1 Zadatak 1 - Opcija 1

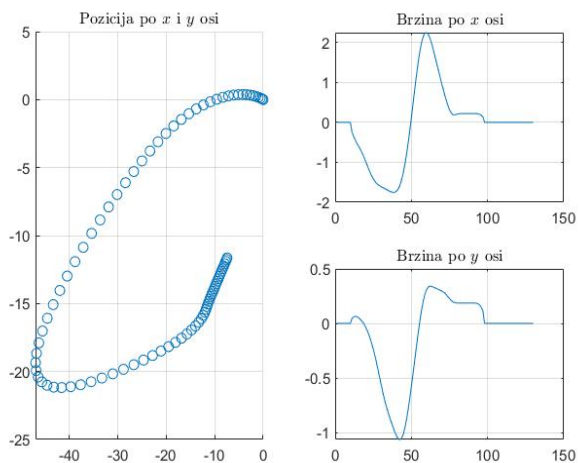
Na sajtu predmeta dostupna je baza zabeleženih brzina po x i y osi, kao i pritisaka zabeleženih prilikom ispisivanja slova na grafičkoj tabli pod nazivom *PO_slova_brzine* (u njoj se nalazi fajl *PO_slova.mat*). Za jedno slovo prva vrsta su brzine po x osi u ekvidistantnim trenucima, druga vrsta brzine po y osi, a treća vrsta pritisak (sve u ekvidistantnim trenucima). Za datu bazu projektovati inovativni sistem za prepoznavanje 10 slova po izboru zasnovan na testiranju hipoteza. Zbog malog broja dostupnih odbiraka za svako slovo, nije potrebna podela na trening i test skup. Uzeti približno jednak broj odbiraka za svako od slova. Nije dozvoljeno koristiti obeležja korišćena na času vežbi ($\max(v_{x||y}) - \min(v_{x||y})$), kao i $\max(v_{x||y})$ i $\min(v_{x||y})$.

- Za jedan primerak svakog od odabranih slova prikazati njegov oblik (poziciju na y osi u zavisnosti od pozicije na x) i njegove brzine po x i y osi. Prokomentarisati povezanost brzine po x i y osi sa načinom ispisivanja slova.
- Rezultate klasifikacije prikazati u obliku matrice konfuzije.
- Opisati projektovani sistem, obrazložiti izbor obeležja i prikazati i prokomentarisati karakteristične primere pravilno i nepravilno klasifikovanih slova (ukoliko ih ima).
- Odabrati dva slova i dva obeležja takva da su odabrana dva slova separabilna u tom prostoru.
- Za slova i obeležja pod d) projektovati parametarski klasifikator po izboru i iscrtati klasifikacionu liniju.

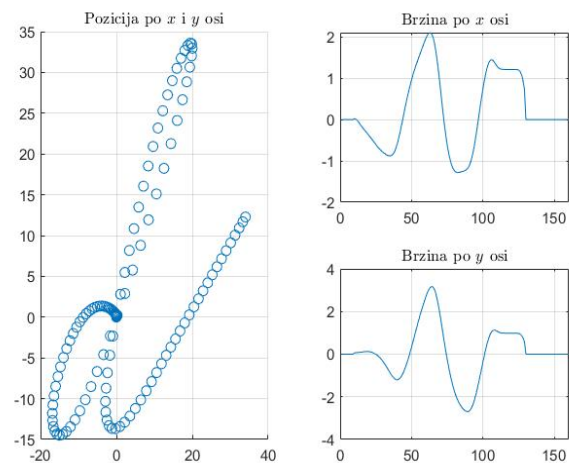
a) Baza slova se sastoji od 20 različitih abecednih slova, gde je svako slovo opisano brzinom po x i y osi, kao i vremenskom zavisnošću pritiska. Odabrao sam prvih 10 slova iz baze *PO_slova.mat* - slova A, B, C, D, E, G, H, I, M i N.



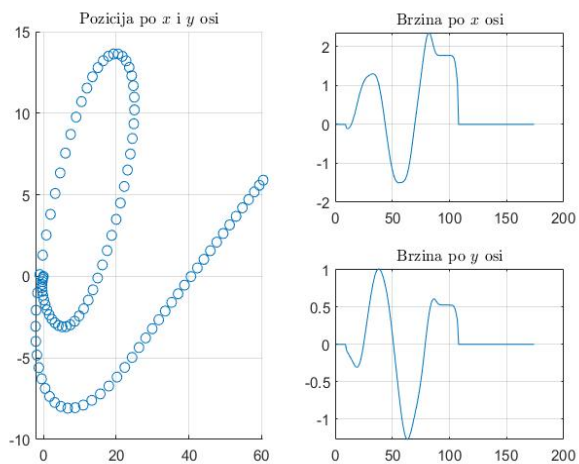
Slovo C



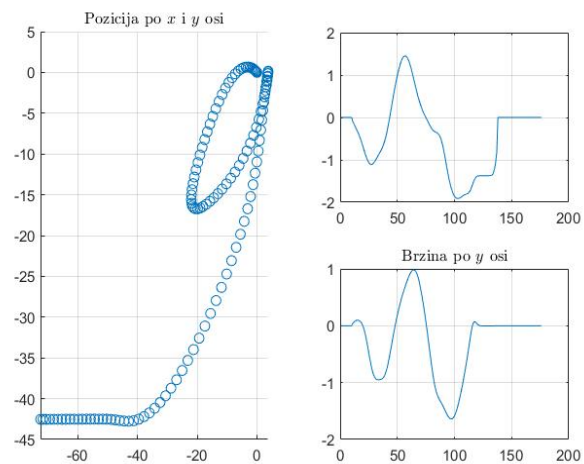
Slovo D



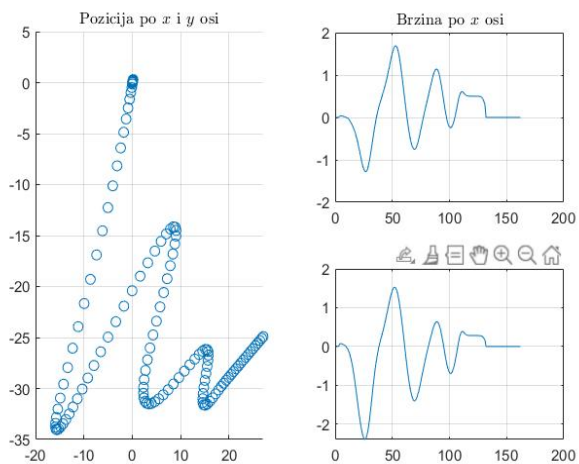
Slovo E



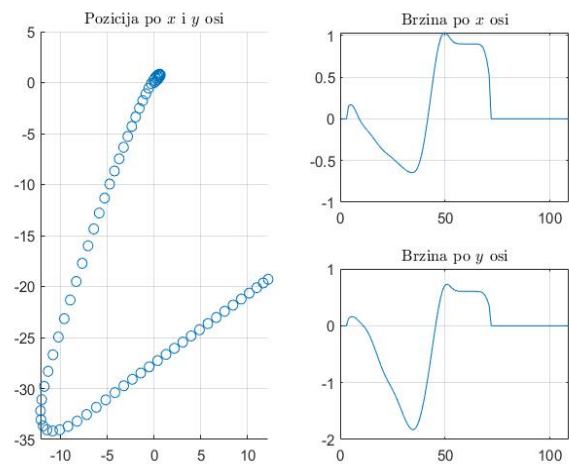
Slovo G

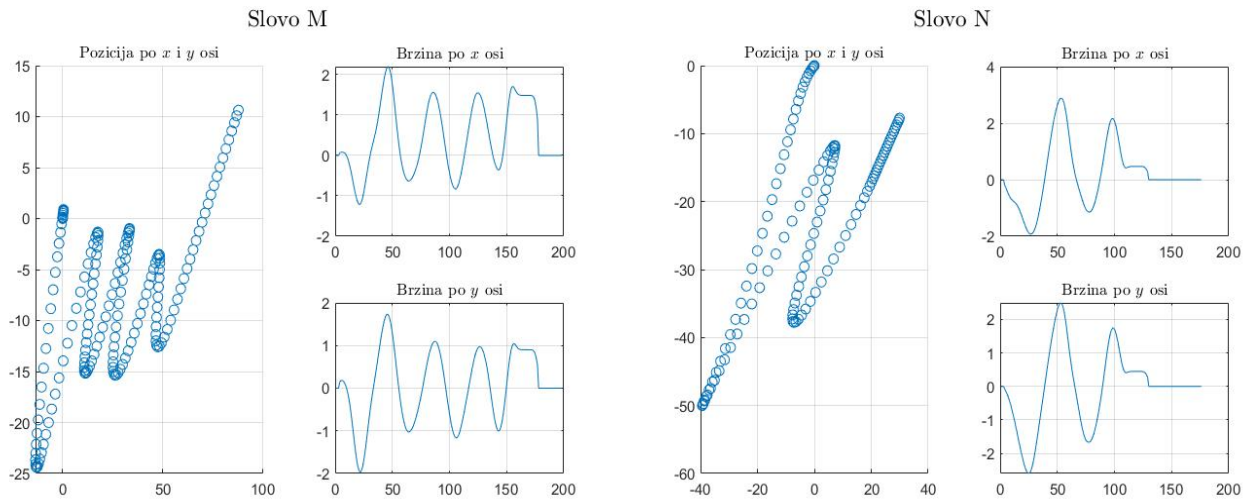


Slovo H



Slovo I





Brzine po x i y osi dosta informativno opisuju način ispisivanja slova. Ako brzina po x osi raste, to znači da prilikom pisanja slova trajektorija pisanja slova je usmerena udesno, dok ako brzina po x osi opada, trajektorija pisanja slova je usmerena ulevo. Analogni zaključak se izvod i za brzinu po y osi, s tim što je tad trajektorija pisanja slova usmerena nagore u slučaju rasta brzine po y osi, a nadole u slučaju opadanja brzine po y osi.

b) Rezultati klasifikacije priloženi su pomoću konfuzione matrice prikazane u nastavku u Tabeli 1.

	A	B	C	D	E	G	H	I	M	N
A	86	0	12	2	0	0	0	0	0	0
B	0	100	0	0	0	0	0	0	0	0
C	11	0	89	0	0	0	0	0	0	0
D	3	0	0	66	0	0	0	0	11	20
E	1	0	3	8	84	0	0	0	2	2
G	0	0	0	0	0	100	0	0	0	0
H	0	0	0	0	1	0	94	0	0	5
I	0	0	0	0	0	0	0	100	0	0
M	0	0	0	11	0	0	1	0	85	3
N	1	0	1	36	6	0	9	0	1	46

Tabela 1: Konfuziona matrica klasifikacije

Uočavamo da sva slova osim slova D (tačnost 66%) i slova M (tačnost 46%) su veoma dobro klasifikovana. Iz konfuzione matrice vidimo da se često slovo N klasifikuje kao slovo D i zbog toga ukupna greška klasifikacije iznosi 15%.

c) Napravljena je posebna funkcija koja ekstrahuje obeležja iz podataka slova, a odabrao sam nekoliko obeležja za koja sam smatrao da mogu doprineti klasifikaciji:

- srednja brzina po x osi;
- srednja brzina po y osi;
- razlika maksimalne i minimalne pozicije po x osi;
- razlika maksimalne i minimalne pozicije po y osi;
- odnos razlika maksimalnih i minimalnih pozicija po x i y osi;
- broj elemenata u donjoj polovini slova;
- razlika maksimalnog i minimalnog pritiska;

Nakon ekstrakcije obeležja za svaki odbirak svakog slova, izvršio sam redukciju dimenzija na bazi međuklasnog rastojanja (engl. *Linear Discriminant Analysis* - **LDA**) sa 7 na 3 dimenzije.

LDA redukcija dimenzija zasniva se na ideji da se oveležja koja treba da se redukuju na neki način spoje sa značajem njihovih informacija, što očuvava separabilnost klasa, ukoliko su klase separabilne. Prvo sam procenio vektore matematičkog očekivanja i kovarijacione matrice za svako od slova pomoću ugrađenih funkcija *mean()* i *cov()*. Zatim sam odredio matricu unutarklasnog rasejanja prema relaciji:

$$S_w = \sum_{j=1}^L \hat{P}_j \hat{\Sigma}_j,$$

gde su \hat{P}_j apriorne verovatnoće pojavljivanja klasa, $\hat{\Sigma}_j$ procenjene kovarijacione matrice za svaku klasu, a broj L ukupan broj klasa (u našem slučaju, $L = 10$). Pošto je za svako slovo odabrano po 100 odbiraka iz baze podataka, apriorne verovatnoće pojavljivanja klasa tada su za svako slovo jednake i iznose $\frac{1}{10}$.

Nakon toga, određuje se vektor združenog matematičkog očekivanja za sve klase zajedno:

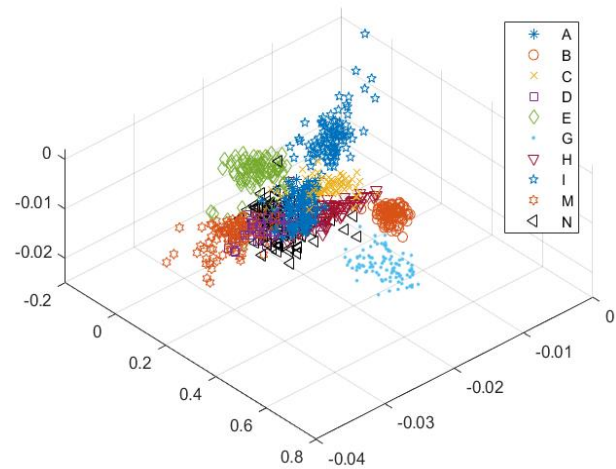
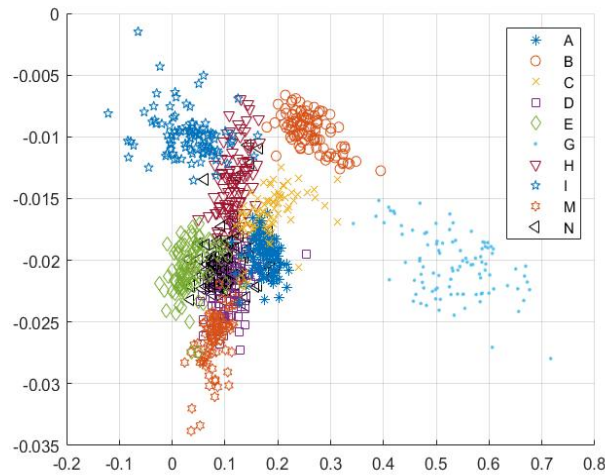
$$M_0 = \sum_{j=1}^L \hat{P}_j \hat{M}_j,$$

gde su \hat{M}_j procenjeni vektori matematičkog očekivanja svake klase.

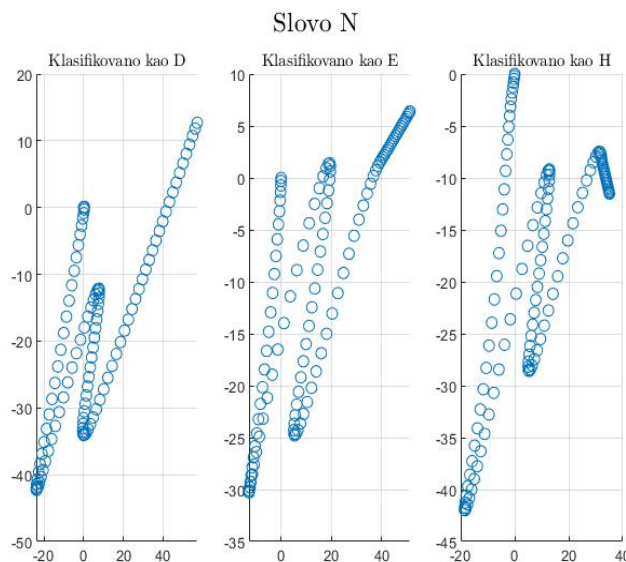
Na osnovu prethodne relacije, definišemo i matricu međuklasnog rasejanja:

$$S_b = \sum_{j=1}^L \hat{P}_j (\hat{M}_j - M_0)(\hat{M}_j - M_0)^T.$$

Tada definišemo matricu $J = (S_w)^{-1} S_b$ i tražimo njene sopstvene vrednosti λ_i i njima odgovarajuće sopstvene vektore Φ_i ($i = 1, 2, \dots, L$). Odabirom tri najveće sopstvene vrednosti određujemo i transformacionu matricu $\Phi = [\Phi_1 \ \Phi_2 \ \Phi_3]$ kojom se vrši dalja redukcija dimenzija.



Na prethodnoj slici uočavamo kako izgledaju raspoređeni odbirci za svako od slova nakon **LDA** redukcije dimenzija na 2, odnosno 3 dimenzije. Uočavamo da neka slova su separabilna (kao što su slova B i G i za njih će biti projektovan linearni klasifikator), dok neka druga slova imaju preklapanja između odbiraka i ona nisu separabilna (kao što su na primer slova N i D), što prouzrokuje grešku u klasifikaciji, a što se između ostalog i vidi u konfuzionoj matrici u Tabeli 1 i ilustrovano na slici ispod.

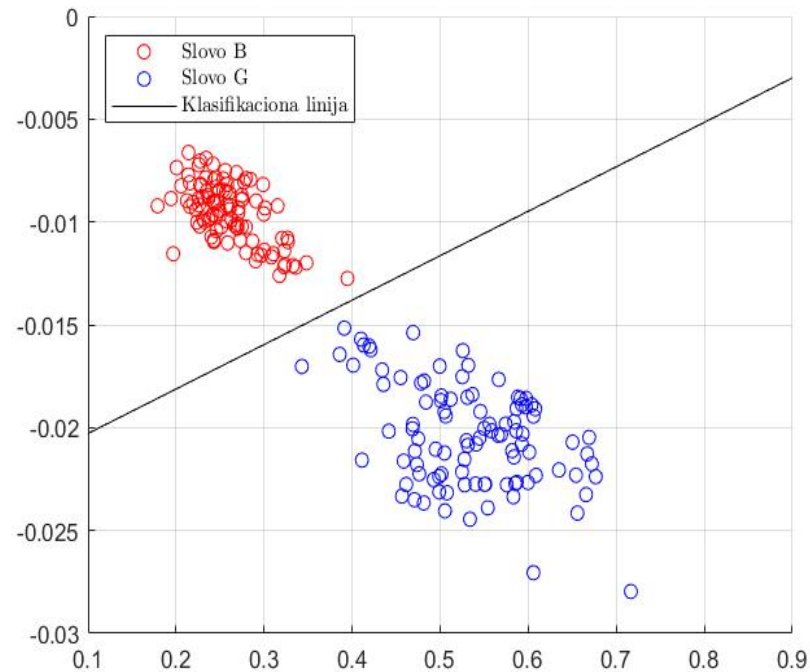


Klasifikacija je izvršena na osnovu testa više hipoteza gde klasifikator ima sledeći oblik:

$$q_k(X) = \max\{q_1(X), q_2(X), \dots, q_L(X)\} \Rightarrow X \in \omega_k,$$

gde su $q_i(X)$ određene relacijom $\frac{\hat{P}_i \hat{f}_i(X)}{\sum_{j=1}^L \hat{P}_j \hat{f}_j(X)}$, a $\hat{f}_j(X)$ su procenjene funkcije gustine verovatnoće svake klase.

d) i e) Na osnovu slike redukcije dimenzija na 2 dimenzije, kao i na osnovu dobijenih rezultata iz konfuzione matrice, vidimo da su klase koje predstavljaju slova B i G separabilne i da su perfektno klasifikovane. S tim u vezi, projektovan je linearni klasifikator preko drugog numeričkog metoda (metod resupstitucije), a rezultati su prikazani na slici ispod.



2 Zadatak 2

Generisati po $N = 500$ odbiraka iz dveju dvodimenzionih bimodalnih klasa:

$$\Omega_1 \sim P_{11}N(M_{11}, \Sigma_{11}) + P_{12}N(M_{12}, \Sigma_{12}),$$

$$\Omega_2 \sim P_{21}N(M_{21}, \Sigma_{21}) + P_{22}N(M_{22}, \Sigma_{22}).$$

Parametre klasa samostalno izabrati.

- Na dijagramu prikazati odbirke.
- Iscrtati kako teorijski izgledaju funkcije gustine verovatnoće za raspodele klasa i uporediti ih sa histogramom generisanih odbiraka.
- Projektovati Bajesov klasifikator minimalne greške i na dijagramu, zajedno sa odbircima, skicirati klasifikacionu liniju. Uporediti grešku klasifikacije konkretnih odbiraka sa teorijskom greškom klasifikacije prve i druge vrste za datu postavku.
- Projektovati klasifikator minimalne cene tako da se više penalizuje pogrešna klasifikacija odbiraka iz prve klase.
- Ponoviti prethodnu tačku za *Neyman-Pearson*-ov klasifikator. Obrazložiti izbor $\epsilon_2 = \epsilon_0$.
- Za klase oblika generisanih u prethodnim tačkama, projektovati *Wald*-ov sekvencijalni test pa skicirati zavisnost broja potrebnih odbiraka od usvojene verovatnoće grešaka prvog, odnosno drugog tipa.

a) Parametri klasa (vektori srednjih vrednosti i kovarijacione matrice klasa) korišćeni u ovom zadatku su:

$$M_{11} = \begin{bmatrix} 6 \\ 8 \end{bmatrix}$$

$$\Sigma_{11} = \begin{bmatrix} 0.7 & -0.5 \\ -0.5 & 0.7 \end{bmatrix}$$

$$M_{12} = \begin{bmatrix} 5 \\ 5 \end{bmatrix}$$

$$\Sigma_{12} = \begin{bmatrix} 0.8 & 0.5 \\ 0.5 & 0.8 \end{bmatrix}$$

$$M_{21} = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$$

$$\Sigma_{21} = \begin{bmatrix} 1.2 & -0.8 \\ -0.8 & 1.2 \end{bmatrix}$$

$$M_{22} = \begin{bmatrix} 0.5 \\ 1.5 \end{bmatrix}$$

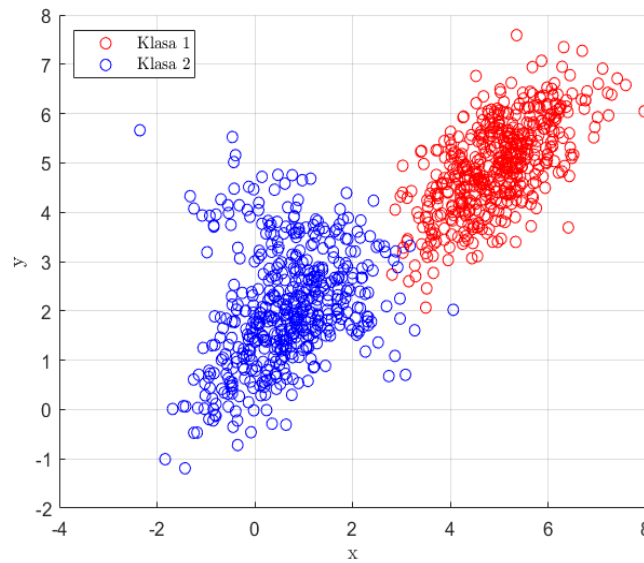
$$\Sigma_{22} = \begin{bmatrix} 0.8 & 0.5 \\ 0.5 & 0.8 \end{bmatrix}$$

$$P_{11} = 0.6$$

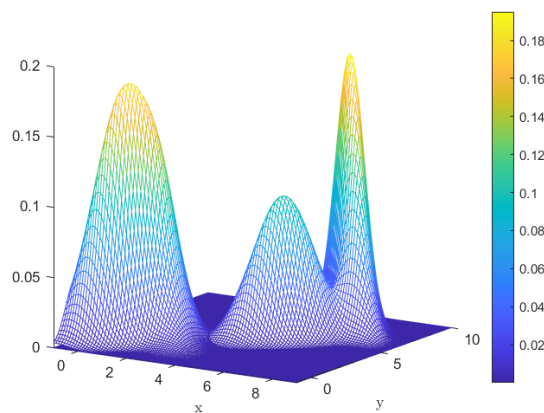
$$P_{12} = 0.4$$

$$P_{21} = 0.3$$

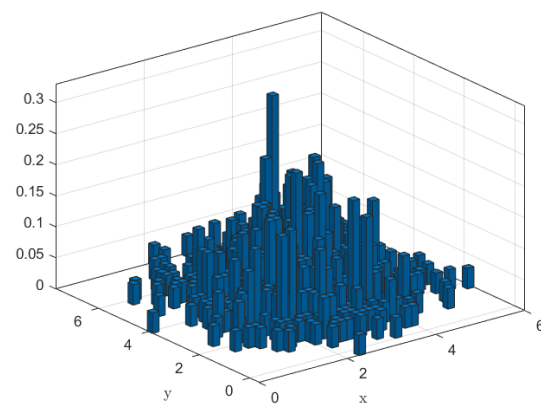
$$P_{22} = 0.7$$

Slika 1: Prikaz odbiraka u xy ravni

b)



(a) Funkcije gustine verovatnoće obe klase



(b) Histogram klase

Slika 2: Prikaz funkcija gustine verovatnoće obe klase i histograma obe klase

Na slici 2 uočavamo da se najveći broj odbiraka grupiše oko vrednosti matematičkih očekivanja koje smo zadali u postavci zadatka. Na primer, vidimo najveće brega oko vrednosti (6,8) i vrednosti (1,3) za prvu odnosno drugu klasu. Kod prve klase jasno se vide dva odvojena brega, dok kod druge se oni ne vide lepo jer su matematička očekivanja jako blizu.

c) Bajesov klasifikator minimalne verovatnoće greške podrazumevano klasifikuje odbirke u dve klase (u opštem slučaju postoji L klasa) tako što određuje aposteriornu verovatnoću pripadanja odbiraka X klasi ω_k na sledeći način:

$$q_k(X) = P(\omega_k|X) = \frac{P_k f_k(X)}{\sum_{i=1}^L P_i f_i(X)}.$$

U slučaju dve klase, klasifikator je oblika:

$$q_1(X) > q_2(X) \Rightarrow X \in \omega_1;$$

$$q_2(X) > q_1(X) \Rightarrow X \in \omega_2.$$

Diskriminaciona kriva $h(X)$ dobija se kao:

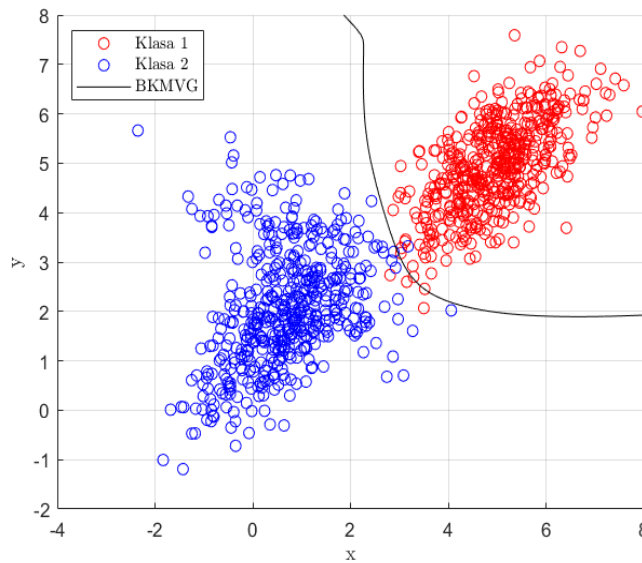
$$h(X) = -\ln \left(\frac{f_1(X)}{f_2(X)} \right).$$

Pravilo zaključivanja sada glasi:

$$h(X) > \ln \left(\frac{P_1}{P_2} \right) \Rightarrow X \in \omega_2;$$

$$h(X) < \ln \left(\frac{P_1}{P_2} \right) \Rightarrow X \in \omega_1.$$

Geometrijski interpretirano, granica odlučivanja klasifikatora predstavlja geometrijsko mesto tačaka koje zadovoljava relaciju $h(X) = 0$. Na slici 3, ova granica je prikazana crnom linijom i predstavlja Bajesov klasifikator minimalne verovatnoće greške.



Slika 3: Prikaz odbiraka i Bajesovog klasifikatora minimalne verovatnoće greške u xy ravni

Dobijeni klasifikator i njegovu grešku klasifikacije, odnosno njegove verovatnoće greške prvog i drugog tipa možemo odrediti tako što na osnovu generisanih odbiraka (čije pripadnosti klasama poznajemo) uz pomoć klasifikatora za svaki odbirak napravimo predikciju, pa ukoliko je predikcija različita od klase kojoj dati odbirak pripada računamo kao grešku klasifikacije. Ovaj pristup je zgodno predstaviti u vidu konfuzione matrice, koja će biti prikazana u nastavku na slici 4.

Grešku klasifikacije možemo odrediti i teorijski tako što ćemo za svaku klasu računati integral funkcije gustine verovatnoće date klase u oblasti one druge klase. Integral sam rešio numerički tako što sam xy ravan izdelio na male kvadrate dimenzije 0.1×0.1 i onda u oblasti klase 2 vrednost funkcije gustine verovatnoće prve klase množio upravo sa tim kvadratom dimenzije 0.1×0.1 i rezultat kumulativno sabirao čime sam odredio verovatnoću greške prvog tipa. Analogno se određuje (teorijska) verovatnoća greške drugog tipa.

```
Konf_matrica =

    497      3
      1   499

Verovatnoca greske 1. tipa: 0.2%
Verovatnoca greske 2. tipa: 0.6%
Teorijska verovatnoca greske 1. tipa: 0.38446%
Teorijska verovatnoca greske 2. tipa: 0.28393%
```

Slika 4: Prikaz konfuzione matrice i odgovarajućih eksperimentalnih i teorijskih verovatnoća grešaka Bajesovog klasifikatora minimalne verovatnoće greške

d) Klasifikator minimalne cene podrazumeva to da se odredi cena koju plaćamo ukoliko pogrešno klasifikujemo neke odbirke, odnosno ako klasifikujemo odbirke iz prve klase u drugu i obrnuto. Cene koje plaćamo prilikom dobre klasifikacije su najčešće 0. U ovom zadatku potrebno je da se više penalizuje pogrešna klasifikacija odbiraka iz prve klase pa sam iz tog razloga odabrao da cena c_{21} 5 puta veća nego cena c_{12} . Pravilo odlučivanja Bajesovog testa minimalne cene glasi:

$$\frac{f_1(X_i)}{f_2(X_i)} > \frac{P_2 c_{12} - c_{22}}{P_1 c_{21} - c_{11}} \Rightarrow X_i \in \omega_1;$$

$$\frac{f_1(X_i)}{f_2(X_i)} < \frac{P_2 c_{12} - c_{22}}{P_1 c_{21} - c_{11}} \Rightarrow X_i \in \omega_2.$$

Bajesov klasifikator minimalne cene prikazan je na slici 5 zelenom linijom.

e) *Neyman-Pearson*-ov test koristi se u situacijama kada greške prvog i drugog tipa imaju različitu težinu (npr. ako je značajnije da greška ϵ_1 bude manja od ϵ_2). Pošto je neophodno ponoviti tačku d), u nastavku greška ϵ_1 ima veću težinu od greške ϵ_2 , odnosno, potrudimo se da grešku ϵ_1 minimizujemo što je više moguće, dok ćemo grešku ϵ_2 držati konstantnom. *Neyman-Pearson*-ov klasifikator dobija se minimizacijom sledećeg kriterijuma:

$$J = \epsilon_1 + \mu(\epsilon_2 - \epsilon_0),$$

gde je μ Lagranžov multiplikator.

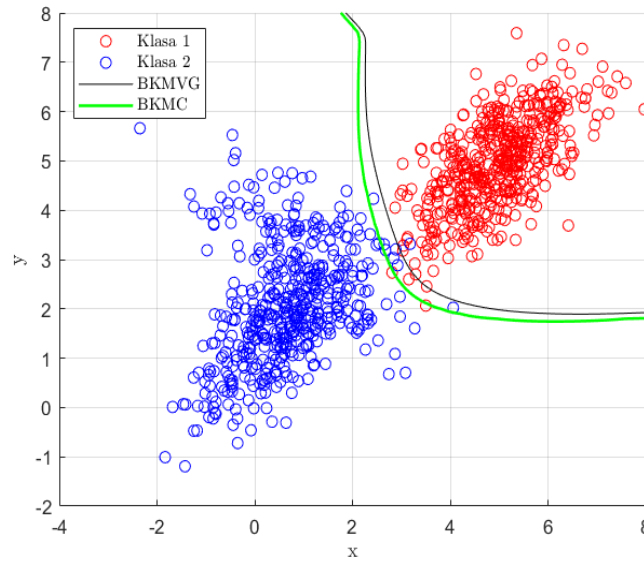
Matematičkom aritmetikom dobijamo da pravilo odlučivanja *Neyman-Pearson*-ovog testa glasi:

$$h(X) = -\ln\left(\frac{f_1(X_i)}{f_2(X_i)}\right) > -\ln(\mu) \Rightarrow X_i \in \omega_2;$$

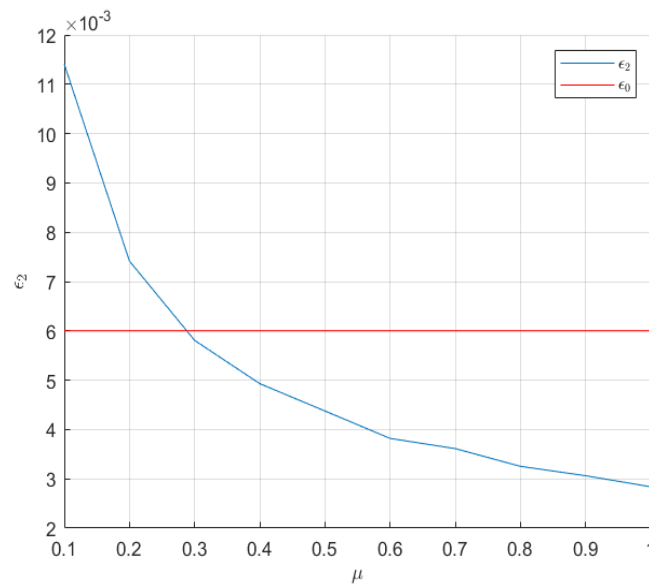
$$h(X) = -\ln\left(\frac{f_1(X_i)}{f_2(X_i)}\right) < -\ln(\mu) \Rightarrow X_i \in \omega_1.$$

Prag μ se, za zadato ϵ_0 određuje iz relacije:

$$\epsilon_2 = \epsilon_0 = \int_{-\infty}^{-\ln(\mu)} f(h|\omega_2)dh.$$

Slika 5: Prikaz odbiraka i Bajesovog klasifikatora minimalne cene u xy ravni

Za parametar ϵ_0 se u najvećem broju slučajeva uzima vrednost koja je jednaka zbiru obe greske ϵ_1 i ϵ_2 (te greške smo dobili na osnovu Bajesovog klasifikatora minimalne verovatnoće greške iz prethodnog dela zadatka) jer na taj način smo sigurni da ukupna greška nikad neće biti veća od te, samim tim minimizujemo vrednost koju može da ima ϵ_2 . Na osnovu grafika zavisnosti greške ϵ_2 od praga μ na slici 6 uočavamo da je dobijena vrednost greške $\epsilon_2 = \epsilon_0$ za $\mu = 0.28$.

Slika 6: Grafik zavisnosti ϵ_2 od praga μ

f) Kada govorimo o *Wald*-ovom testu, podrazumevamo to da odбирke dobijamo sekvencijalno (jedan po jedan), a ne da su nam svi poznati na početku. Takođe, smatramo da su svi odbirci iz iste klase. Klasifikator ima sledeći oblik:

$$S_m \in \begin{cases} (-\infty, a), & X_m \in \omega_1 \\ [a, b], & \text{uzmi sledeći odbirak } X_{m+1} \\ (b, \infty), & X_m \in \omega_2 \end{cases}$$

pri čemu je:

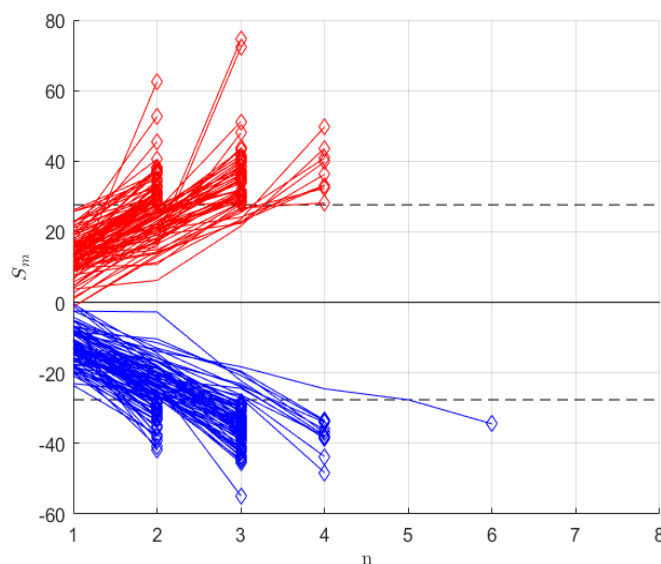
$$S_m = \sum_{i=1}^m h(X_i) = \sum_{i=1}^m -\ln \left(\frac{f_1(X_i)}{f_2(X_i)} \right).$$

Parametre a i b određujemo na osnovu željenih vrednosti vrednosti grešaka prve i druge vrste ϵ_1 i ϵ_2 preko sledećih relacija:

$$a = -\ln \left(\frac{1 - \epsilon_1}{\epsilon_2} \right); \quad b = -\ln \left(\frac{\epsilon_1}{1 - \epsilon_2} \right).$$

Za ovakav odabir parametara a i b , odnosno grešaka ϵ_1 i ϵ_2 , *Wald*-ov test minimizira broj potrebnih odbiraka za donošenje odluke.

Na slici 7 demonstriran je rad *Wald*-ovog testa. Greške ϵ_1 i ϵ_2 su jednake i iznose 10^{-12} . Ukoliko bi greške bile veće, algoritam bi donosio odluku nakon svakog odbirka pojedinačno pošto su klase separabilne, pa su ove vrednosti odabrane kako bi se rezultat jasno ilustrovao. Za vrednosti a i b su dobijene vrednosti ± 27.6310 . Plava boja predstavlja odbirke prve klase, a crvena druge. Na osnovu grafika se vidi da je najčešće potrebno 4 odbirka kako bi se donela ispravna odluka.



Slika 7: *Wald*-ov sekvencijalni test

Srednji broj odbiraka potrebnih da se donese odluka da je odbirak iz prve ili druge klase se računa po sledećim relacijama:

$$m_1 = \frac{-\ln\left(\frac{1-\epsilon_1}{\epsilon_2}\right)(1-\epsilon_1) - \ln\left(\frac{\epsilon_1}{1-\epsilon_2}\right)\epsilon_1}{\eta_1},$$

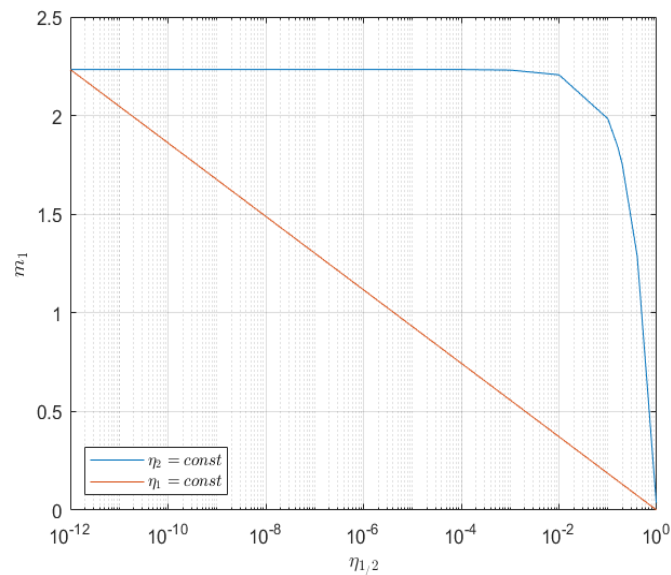
$$m_2 = \frac{-\ln\left(\frac{1-\epsilon_1}{\epsilon_2}\right)\epsilon_2 - \ln\left(\frac{\epsilon_1}{1-\epsilon_2}\right)(1-\epsilon_2)}{\eta_2},$$

pri čemu su:

$$\eta_1 = E(h|\omega_1)$$

$$\eta_2 = E(h|\omega_2).$$

Na slici 8 prikazan je grafik zavisnosti potrebe broja odbiraka prve klase u zavisnosti od grešaka ϵ_1 i ϵ_2 . Možemo uočiti da ϵ_1 utiče na m_1 samo kada ima velike vrednosti, dok sa porastom ϵ_2 vrednost m_1 linearno opada.



Slika 8: Zavisnost prosečnog broja potrebnih odbiraka prve klase u zavisnosti od vrednosti grešaka ϵ_1 i ϵ_2

3 Zadatak 3 - Opcija 1

1. Generisati tri klase dvodimenzionalnih oblika. Izabrati funkciju gustine verovatnoće oblika tako da klase budu linearno separabilne.

a) Za tako generisane oblike izvršiti projektovanje linearnog klasifikatora jednom od tri iterativne procedure. Rezultate prikazati u obliku matrice konfuzije. Detaljno opisati postupak klasifikacije.

b) Ponoviti prethodni postupak korišćenjem metode željenog izlaza. Analizirati uticaj elemenata u matrici željenih izlaza na konačnu formu linearnog klasifikatora.

2. Generisati dve klase dvodimenzionalnih oblika koje jesu separabilne, ali ne linearno, pa isprojektovati kvadratni klasifikator metodom po želji.

1a) Generisane su tri klase dvodimenzionalnih oblika sa sledećim statističkim parametrima:

$$M_1 = \begin{bmatrix} -8 \\ -8 \end{bmatrix} \quad \Sigma_1 = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix}$$

$$M_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \Sigma_2 = \begin{bmatrix} 1 & 0.2 \\ 0.2 & 1 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 8 \\ 0 \end{bmatrix} \quad \Sigma_3 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$$

Parametarski klasifikator se sastoji od vektora oblika $h(X) = V^T X + v_0 = 0$ koji predstavlja granicu odlučivanja između dve klase čiji su vektori matematičkog očekivanja M_i i M_j , kao i odgovarajuće kovarijacione matrice Σ_i i Σ_j se određuje tako da se zadovolje sledeće relacije:

$$V = (s\Sigma_i + (1-s)\Sigma_j)^{-1}(M_j - M_i), \quad (1)$$

$$s = \frac{\frac{-(V^T M_i + v_0)}{\sigma_i^2}}{\frac{-(V^T M_i + v_0)}{\sigma_i^2} + \frac{V^T M_j + v_0}{\sigma_j^2}}, \quad (2)$$

$$v_0 = -\frac{s\sigma_i^2 V^T M_j + (1-s)\sigma_j^2 V^T M_i}{s\sigma_i^2 + (1-s)\sigma_j^2}, \quad (3)$$

gde su σ_i^2 i σ_j^2 određeni izrazima:

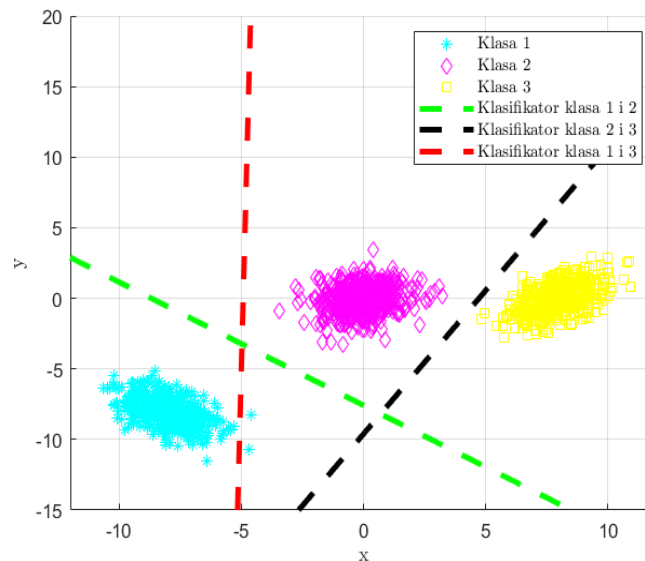
$$\sigma_i^2 = V^T \Sigma_i V, \quad \sigma_j^2 = V^T \Sigma_j V. \quad (4)$$

S obzirom da postoje tri klase, neophodno je projektovati tri linearna klasifikatora za svake dve klase, što je učinjeno trećom iterativnom procedurom (*holdout* metod) koji se zasniva na ideji da se ulazni skup podataka podeli na dva skupa - obučavajući i testirajući u odnosu 75:25. Zatim se izvršava sledeća iterativna procedura na obučavajućem skupu:

- korak s se menja u intervalu od 0 do 1 sa malim korakom (u mom slučaju korak iznosi 0.001);
- izračuna se V na osnovu jednačine (1);
- zatim se izračunavaju σ_i^2 i σ_j^2 pomoću jednačine (4);

- nakon toga se odrede predikcije $y_k^i = V^T X_k^{(i)}$ i $y_k^j = V^T X_k^{(j)}$ (gde je $X_k^{(i)}$ k -ti odbirak iz i -te klase, a $X_k^{(j)}$ k -ti odbirak iz j -te klase testirajućeg skupa) za sve raspoložive odbirke i smeštaju u matrice Y_i i Y_j ;
- matrice Y_i i Y_j se sortiraju rastuće i za svaka dva susedna elementa matrica traži se v_0 između njih, kao i broj grešaka;
- za s za koje je broj grešaka najmanji, određuje se iz jednačine (1) V_{opt} , a iz jednačine (3) v_{opt} .

Rezultati ovakve klasifikacije prikazani su na slici 9.



Slika 9: Linearna parametarska klasifikacija u xy ravni - *holdout* metod

Konfuzione matrice kao rezultat linearne parametarske klasifikacije tri klase prikazana je na slici 10.

Konfuzionna matrica - klase 1 i 2

```
126    0
  0   126
```

Konfuzionna matrica - klase 2 i 3

```
126    0
  0   126
```

Konfuzionna matrica - klase 1 i 3

```
125    1
  0   126
```

Slika 10: Konfuzione matrice linearne parametarske klasifikacije - *holdout* metod

Klase su dosta separabilne, samim tim su i greške klasifikacije malobrojne, što pokazuje prethodna slika.

1b) Kod metoda željenog izlaza, diskriminaciona kriva linearnog klasifikatora se predstavlja u matricnom obliku na sledeći način:

$$h(X) = W^T X,$$

gde je $W^T = [v_0 \ V]^T$, a vektor Z se formira na sledeći način:

$$Z = [-1 \ -X]^T, \quad X \in \omega_1;$$

$$Z = [1 \ X]^T, \quad X \in \omega_2.$$

Formirajući takav vektor Z , klasifikator dobija oblik:

$$h(Z) = W^T Z = \sum_{i=1}^N w_i z_i > 0.$$

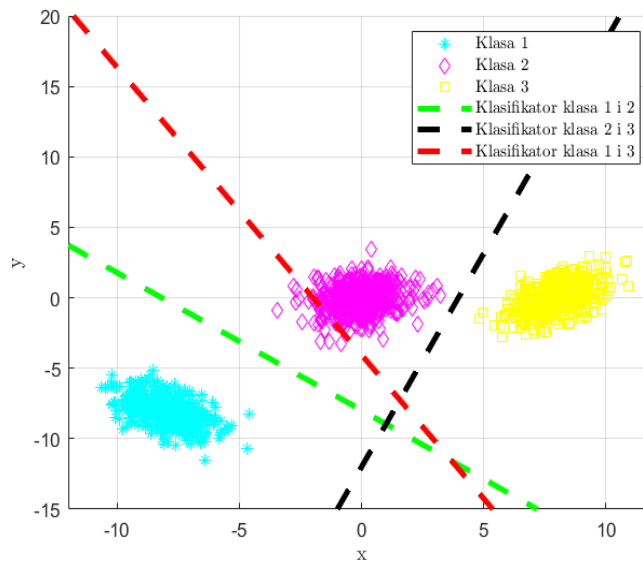
Parametri klasifikatora određuju se minimizacijom kriterijuma koji predstavlja usrednjenu sumu kvadrata dobijenih od željenih izlaza:

$$\bar{\epsilon}^2 = \frac{1}{N} \sum_{i=1}^N (W^T Z_i - \gamma(Z_i))^2.$$

Formiranjem matrica $U = [Z_1, Z_2, \dots, Z_N]$ i $\Gamma = [\gamma(Z_1), \gamma(Z_2), \dots, \gamma(Z_N)]^T$, vektor sa nepoznatim parametrima linearnog klasifikatora na bazi željog izlaza W se određuje preko relacije:

$$W = (UU^T)^{-1}U\Gamma.$$

Rezultati ovakve klasifikacije prikazani su u xy ravni na slici 11



Slika 11: Linearna parametarska klasifikacija u xy ravni - metod željenog izlaza

Konfuzione matrice kao rezultat linearne parametarske klasifikacije tri klase prikazana je na slici 12.

```
Konfuzionna matrica - metod zeljenog izlaza - klase 1 i 2
500    0
  0   500

Konfuzionna matrica - metod zeljenog izlaza - klase 2 i 3
500    0
  0   500

Konfuzionna matrica - metod zeljenog izlaza - klase 1 i 3
500    0
  0   500
```

Slika 12: Konfuzione matrice linearne parametarske klasifikacije - metod željenog izlaza

Ovakva klasifikacija je uspešna stoprocentno sa istim podacima, jer su klase separabilne. Važna napomena je da je u ovoj tački kod izbora matrice željenih izlaza Γ odabrana jedinična matrica, odnosno, sve su klase jednako prioritete. Množenjem nekih elemenata matrice željenog izlaza koji odgovaraju vektorima Z iz određene klase vrednošću većom od 1 znači da se vrši favorizacija te klase u odnosu na drugu klasu, što znači da bi u tom slučaju jednakoprioritetna klasifikaciona linija bila samo pomerena od prve klase, pošto je prva klasa od većeg interesa i bitnije nam je da ne pogrešimo u klasifikaciji kada je prva klasa u pitanju.

2. Diskriminaciona kriva kvadratnog klasifikatora je sledećeg oblika:

$$h(X) = X^T Q X + V^T X + v_0 = 0,$$

gde su:

$$Q = \begin{bmatrix} q_{11} & \dots & q_{1N} \\ \vdots & \ddots & \vdots \\ q_{N1} & \dots & q_{NN} \end{bmatrix}, \quad V^T = [v_1 \ v_2 \ \dots \ v_N].$$

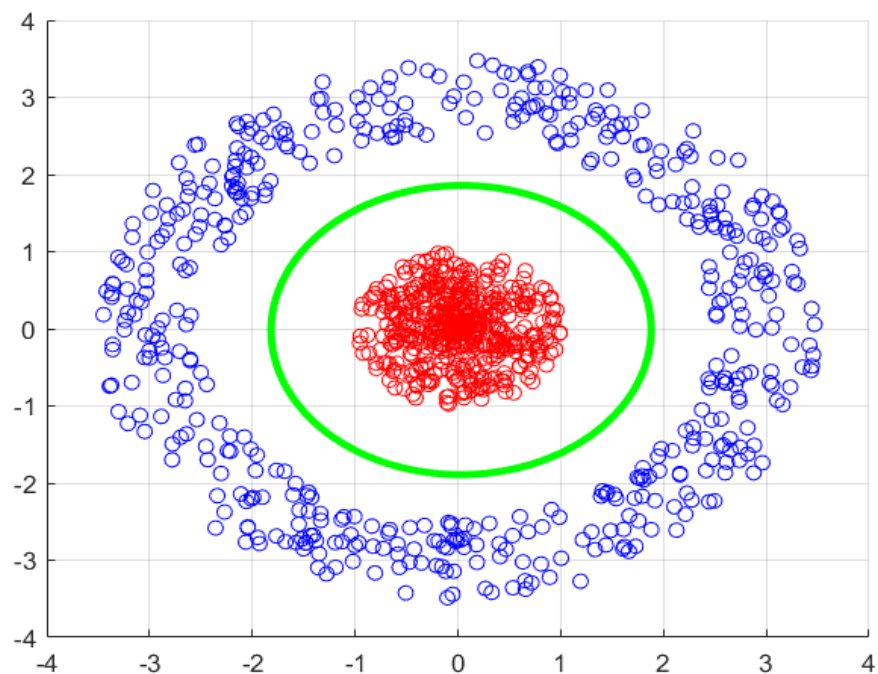
Kako bi se primenio metod željenih izlaza, neophodno je svesti kvadratni klasifikator na linearni, što se postiže sledećim izrazima:

$$h(Y) = D^T Y + v_0,$$

gde su:

$$D^T = [v_1 \ \dots \ v_N \ q_{11} \ (q_{12} + q_{21}) \ \dots \ q_{NN}], \quad Y^T = [X_1 \ \dots \ X_N \ X_1^2 \ X_1 X_2 \ \dots \ X_N^2].$$

Nakon ovakve transformacije, mogu da se primene svi metodi kao i za linearni klasifikator. U nastavku na slici 13 su prikazani rezultati kvadratne klasifikacije dve nelinearno separabilne klase metodom željenog izlaza, ali kada je favorizovana druga klasa (odbirci plave boje). To je postignuto tako što je za svako Z_i koje odgovara drugoj klasi dodeljen željeni izlaz $\gamma(Z_i) = 1.7$, dok je za prvu klasu (odbirci crvene boje) $\gamma(Z_i) = 1$. Klasifikaciona linija (zelena linija) je na taj način pomerena od druge klase.



Slika 13: Kvadratna parametarska klasifikacija u xy ravni - metod željenog izlaza

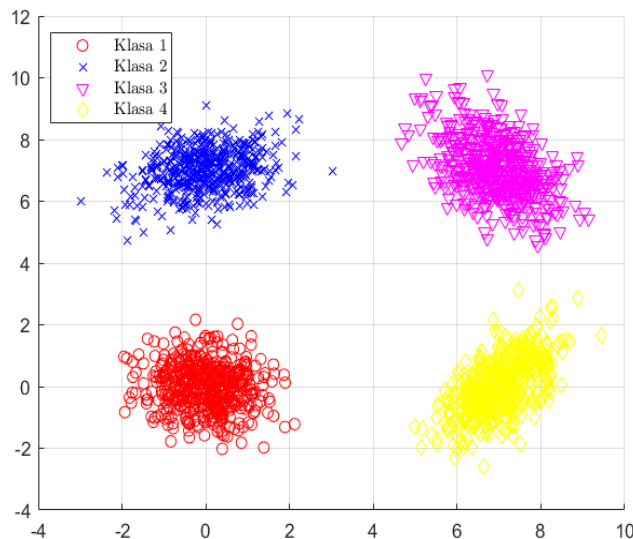
4 Zadatak 4

1. Generisati po $N = 500$ dvodimenzionih odbiraka iz četiri klase koje će biti linearno separabilne. Preporuka je da to budu Gausovski raspodeljeni dvodimenzioni oblici. Izabrati jednu od metoda za klasterizaciju (c mean metod, metod kvadratne dekompozicije) i primeniti je na formirane uzorke klase. Izvršiti analizu osetljivosti izabranog algoritma na početnu klasterizaciju kao i srednji broj potrebnih iteracija. Takođe izvršiti analize slučaja kada se apriorno ne poznaje broj klase.

2. Na odbircima iz prethodne tačke izabrati jednu od metoda klasterizacije (metod maksimalne verodostojnosti ili metod grana i granica) i primeniti je na formirane uzorke klase. Izvršiti analizu osetljivosti izabranog algoritma na početnu klasterizaciju kao i srednji broj potrebnih iteracija. Takođe izvršiti analize slučaja kada se apriorno ne poznaje broj klase.

3. Generisati po $N = 500$ dvodimenzionih odbiraka iz dve klase koje su nelinearno separabilne. Izabrati jednu od metoda za klasterizaciju koje su primenjive za nelinearno separabilne klase (metod kvadratne dekompozicije ili metod maksimalne verodostojnosti) i ponoviti analizu iz prethodnih tačaka.

1. Na slici su prikazane četiri linearno separabilne Gausovski raspodeljene klase. Centri klase su udaljeni kako bismo postigli što bolju separabilnost, tj. kako ne bi došlo do pojave *outlier*-a i kako bismo brže izvršili proces klasterizacije.

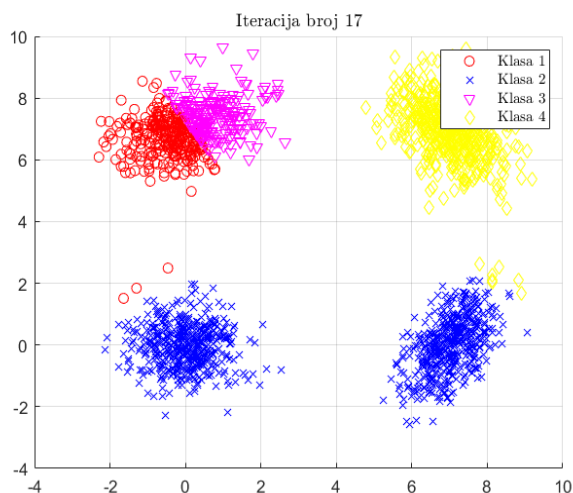


Slika 14: Prikaz odbiraka četiri klase u xy ravni

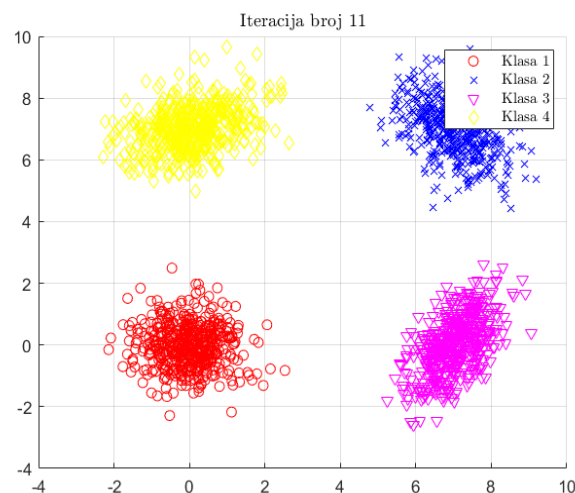
Prvi metod klasterizacije koji se razmatra je c – mean klasterizacija, a njen algoritam je sledeći:

- Izabere se inicijalna klasterizacija i izračunaju se vektori matematičkog očekivanja $M_1^{(0)}, M_2^{(0)}, \dots, M_L^{(0)}$;
- U l -toj iteraciji, na osnovu izračunatih $M_1^{(l)}, M_2^{(l)}, \dots, M_L^{(l)}$ za svaki odbirak X se vrši reklasifikacija odbirka u klasu koja odgovara najbližem vektoru srednjih vrednosti $M_j^{(l)}$ po Euklidskom rastojanju;
- Ako je barem jedan odbirak X reklasifikovan u l -toj iteraciji, ulazi se u novu $l + 1$ -u iteraciju i vraćamo se na drugi korak;
- Ukoliko u tekućoj iteraciji nijedan odbirak X nije reklasifikovan, algoritam se završava.

Za 100 iteracija sa različitim nasumičnim početnim klasterizacijama, dobija se da je prosečan broj iteracija 4.29, gde je najmanje trebalo 2, a najviše 17 iteracija, što rezultuje završetkom u lokalnom minimumu i tada finalna reklasifikacija ne daje dobar rezultat. Ovako loša klasterizacija prikazana je na slici 15(a). Ovaj problem se može prevazići tako što ćemo na početku u svaku od klasa ubaciti npr. 20 posto odbiraka koji stvarno pripadaju svakoj od klasa, a ostatak slučajno raspodeliti. Algoritam će vrlo brzo stići do pravog rešenja, jer na su centri klastera u startu dosta dobro određeni.



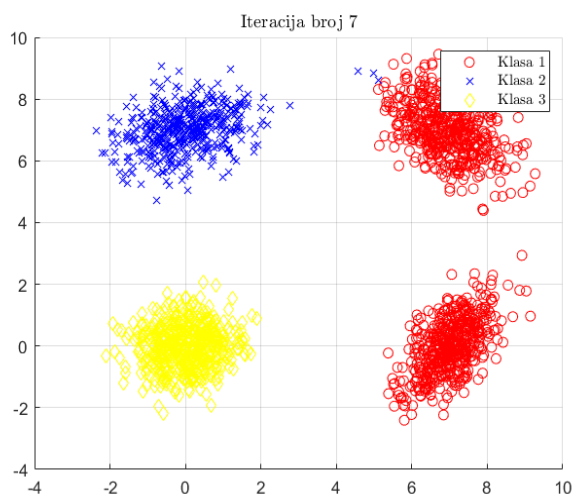
(a) Prikaz završetka klasterizacije u lokalnom minimumu



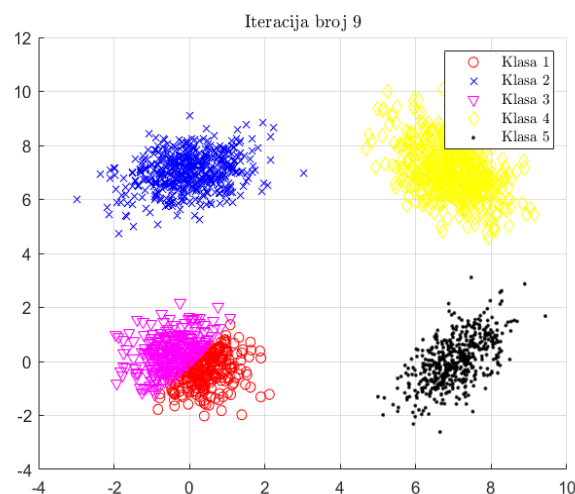
(b) Prikaz završetka klasterizacije sa zamenom klastera

Slika 15: Prikaz neispravnog rada algoritma *c – mean* klasterizacije

Takođe, u zavisnosti od početne klasterizacije, problem je predstavljalo to što se dešavalo da neki klaster postane prazan. Ovaj problem je rešen tako što je pronađen klaster sa najviše elemenata i polovina njegovih semplava je prebačena u prazan klaster, a druga polovina je ostala u originalnom klasteru, čime je omogućeno da se nastavi dalja klasterizacija sa istim brojem unapred željenih klasa. U tom slučaju može da se desi da finalna, dobra klasterizacija zameni mesta klasterima slike 14, što je prikazano na slici 15(b).



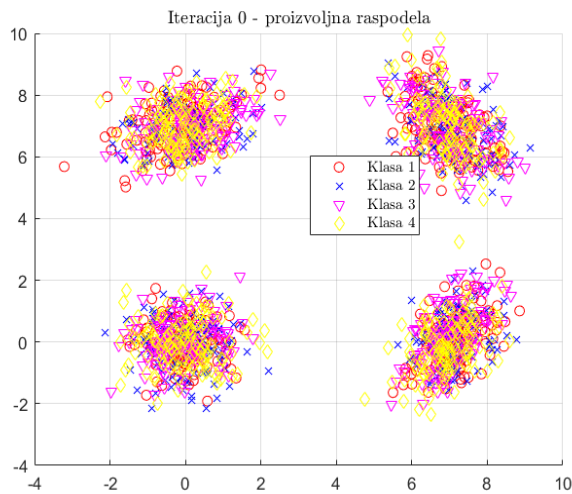
(a) Prikaz završetka klasterizacije u slučaju 3 klase



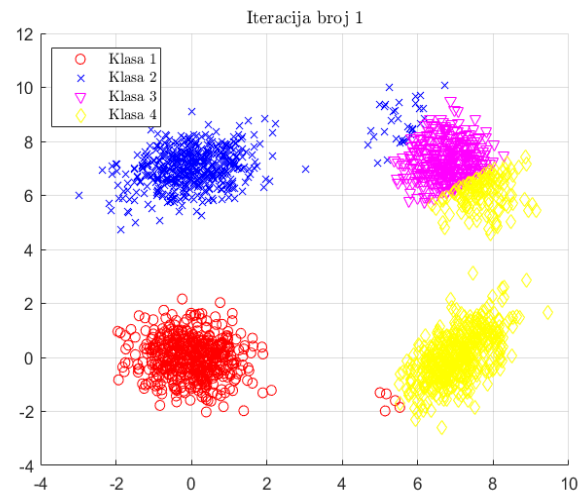
(b) Prikaz završetka klasterizacije u slučaju 5 klasa

Slika 16: Prikaz rada algoritma *c – mean* klasterizacije kada je broj klasa 3 i 5

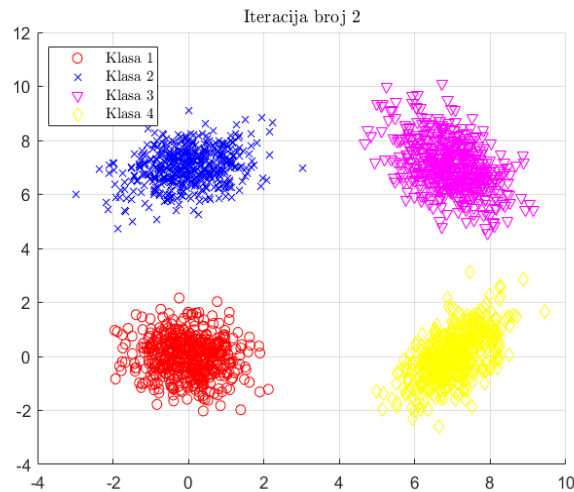
Na slici 16(a) i 16(b) su prikazane finalne klasterizacije kada je promenjen broj klastera na 3 i 5 respektivno. Prilikom klasterizacije, ukoliko nam apriorno nije poznat broj klasa, dolazi do pogrešnog klasterizovanja. U slučaju manjeg broja klasa, algoritam će spojiti dve klase u jednu, dok u slučaju više klasa doći će do podele neke od klasa na dve klase.



(a) Početna klasterizacija



(b) Nakon prve iteracije



(c) Nakon druge iteracije

Slika 17: Prikaz ispravnog rada algoritma *c – mean* klasterizacije

Na slici 17 prikazan je rad algoritma *c – mean* klasterizacije koji dobro izvrši reklasifikaciju nakon dve iteracije. Što su klase separabilnije i što je manji broj *outlier*-a, to je ovaj algoritam tačniji i brži. Algoritam je veoma osetljiv na početnu klasterizaciju koji je glavni uzrok problema. Apriorno poznavanje broja klasa u ovom slučaju je neophodno.

2. Primena metoda maksimalne verodostojnosti pretpostavlja da su funkcije gustine verovatnoća oblika iz pojedinih klasa normalno raspodeljene i da je stoga združena funkcija gustine verovatnoće svih oblika zapravo mešavina (zbir) Gausovih raspodela. Tada se problem može formulisati na sledeći način: Na osnovu raspoloživih oblika potrebno je estimirati združenu funkciju gustine verovatnoće u formi:

$$f(X) = \sum_{i=1}^L P_i f_i(X); \quad f_i(X) \sim N(M_i, \Sigma_i); \quad i = 1, \dots, L.$$

Primena metoda maksimalne verodostojnosti na ovaj problem se svodi na maksimizaciju funkcije $\prod_{j=1}^N f(X_j)$, po parametrima P_i , M_i , Σ_i ($i = 1, \dots, L$). Optimizacijom kriterijuma dobijaju se relacije:

$$q_i(X_j) = \frac{P_i f_i(X_j)}{f(X_j)};$$

$$P_i = \frac{1}{N} \sum_{j=1}^N q_i(X_j);$$

$$M_i = \frac{1}{N_i} \sum_{j=1}^N q_i(X_j) X_j;$$

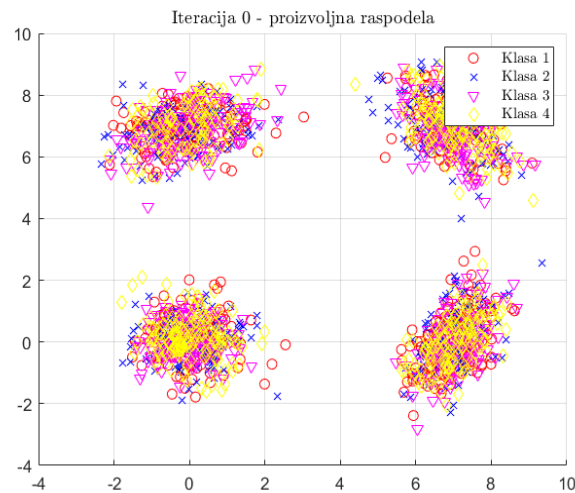
$$\Sigma_i = \frac{1}{N_i} \sum_{j=1}^N q_i(X_j) (X_j - M_i)(X_j - M_i)^T;$$

$$N_i = P_i N.$$

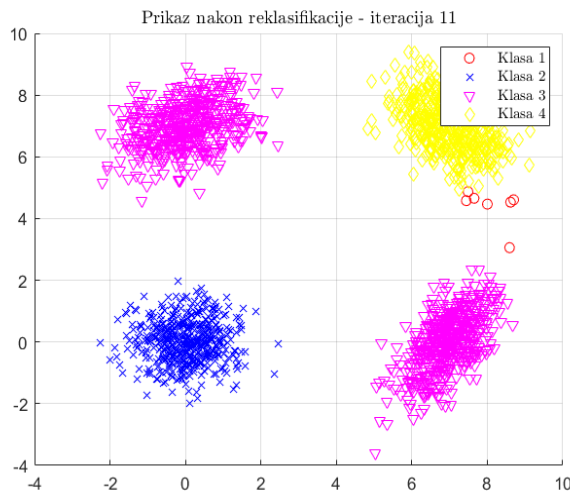
Kako parametri P_i , M_i i Σ_i zavise od verovatnoće q_i i obrnuto, rešenje se ne može eksplicitno odrediti, stoga se koristi iterativni metod prikazan u nastavku:

- Izabere se početna klasterizacija Ω_0 i na osnovu nje se izračunaju P_i , M_i i Σ_i ($i = 1, \dots, L$);
- Na osnovu parametara u l -toj iteraciji $P_i^{(l)}$, $M_i^{(l)}$ i $\Sigma_i^{(l)}$, kao i na osnovu verovatnoće $q_i^{(l)}$, izračunaju se parametri za novu iteraciju $P_i^{(l+1)}$, $M_i^{(l+1)}$ i $\Sigma_i^{(l+1)}$ i na osnovu njih nove verovatnoće $q_i(X_j)^{(l+1)} = \frac{P_i^{(l+1)} f_i^{(l+1)}(X_j)}{\sum_{k=1}^L P_k^{(l+1)} f_k^{(l+1)}(X_j)}$;
- Ako je $q_i(X_j)^{(l)} = q_i(X_j)^{(l+1)}$ za ($\forall i = 1, \dots, L$) i za ($\forall j = 1, \dots, N$), algoritam se završava, inače se vraćamo na drugi korak.

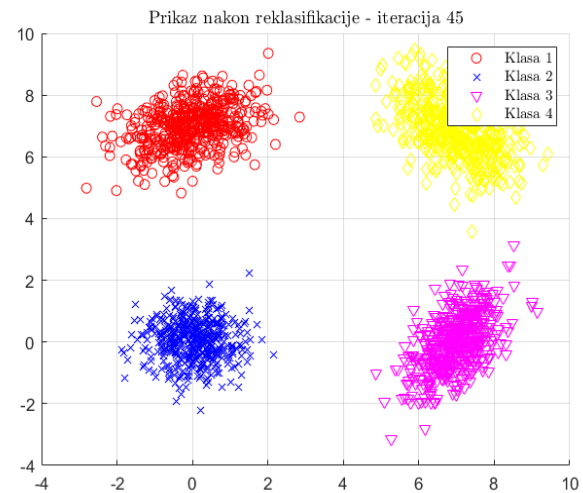
Slično kao i kod *c - mean* klasterizacije, rezultat klasterizacije dosta zavisi od početne klasterizacije. Ukoliko izvršimo početnu klasterizaciju na potpuno slučajan način, u najvećem broju slučajeva ćemo dobiti loše rezultate, i najčešće se dešava da jedan klaster ostane potpuno prazan ili u neki klaster završe odbirci koji ne bi trebalo da završe u tom klasteru (slika 18(b)). Međutim, ukoliko drugačije izvršimo početnu klasterizaciju, tako što ćemo veštački ubaciti po 20 posto odbiraka u svaku od klasa iz klasa kojim odbirci zapravo pripadaju, rezultati će biti jako dobri (slika 18(c)).



(a) Početna klasterizacija



(b) Loša klasterizacija



(c) Dobra klasterizacija

Slika 18: Prikaz neispravnog i ispravnog rada algoritma metoda maksimalne verodostojnosti

Algoritam je veoma osetljiv na početnu klasterizaciju, a dobijen je prosečan broj iteracija od 22.723 iteracija koje su neophodne za finalnu reklasifikaciju onda kada je početna klasterizacija slučajna, i tada ovaj algoritam ne konvergira ka tačnom rešenju. Kada se u početnoj klasterizaciji podesi određen broj odbiraka u klase kojima stvarno pripadaju prosečan broj iteracija je 10.674, pri čemu je i validnost reklasifikacija sada drastično veća.

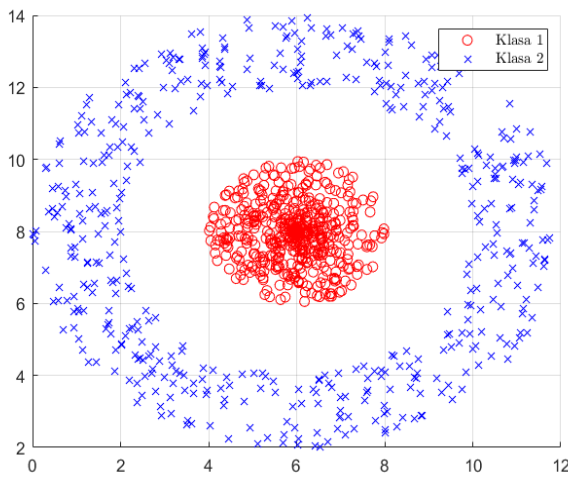
Algoritam je u slučaju veoma osetljiv i na apriorno znanje o broju klasa, čak i kada početna klasterizacija nije slučajna, algoritam ne konvergira ka tačnom rešenju.

U odnosu na $c - mean$, algoritam je značajno sporiji.

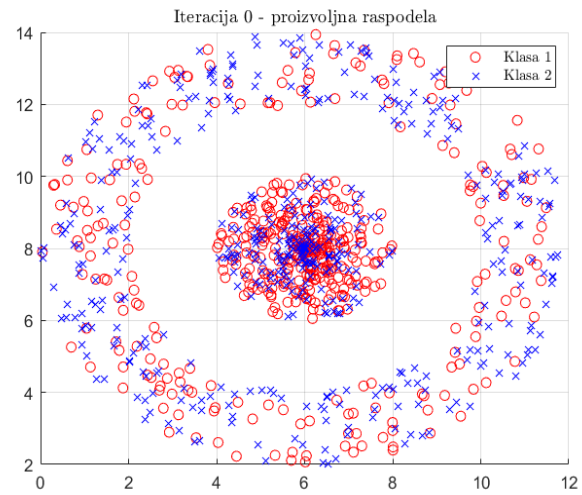
3. U ovom delu zadatka biće demonstriran metod normalne (kvadratne) dekompozicije. Ovo je iterativni postupak sličan *c-mean* metodu, a razlika je u izboru kriterijumske funkcije, koja sada postaje minimizacija statističkog rastojanja odbirka u odnosu na centar klastera, dok je kod *c-mean* metoda kriterijumska funkcija bila minimizacija Euklidskog rastojanja svakog odbirka od najbliže klase. Kriterijumska funkcija je određena izrazom:

$$J = \frac{1}{2}(X_i - M_j^{(l)})^T(\Sigma_j^{(l)})^{-1}(X_i - M_j^{(l)}) + \frac{1}{2} \ln |\Sigma_j^{(l)}| - \ln P_j^{(l)}; \quad j = 1, \dots, L.$$

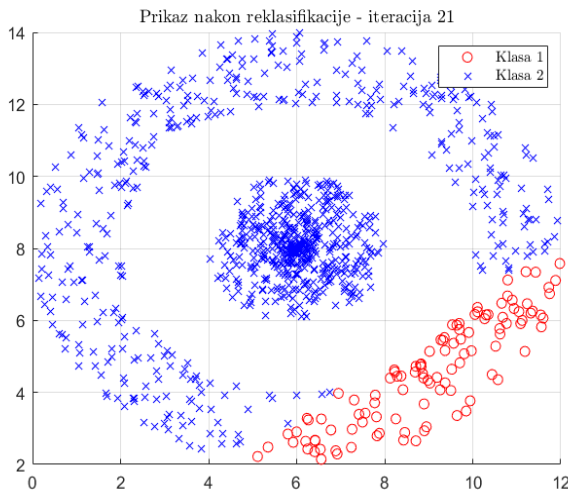
U ovom izrazu vrednost L predstavlja broj klastera, $P_j^{(l)}$ predstavlja verovatnoću da odbirak dolazi iz klase j , $M_j^{(l)}$ srednju vrednost klase j , i $\Sigma_j^{(l)}$ kovarijacionu matricu klase j u iteraciji l . Ovakav izraz za kriterijum J nam govori da je u pitanju klasifikator drugog reda, odnosno, omogućava nam da izvršimo klasterizaciju nelinearno separabilnih klastera, za razliku od *c-mean* metoda koji nam to nije omogućavao, jer je klasifikaciona kriva bila kriva prvog reda.



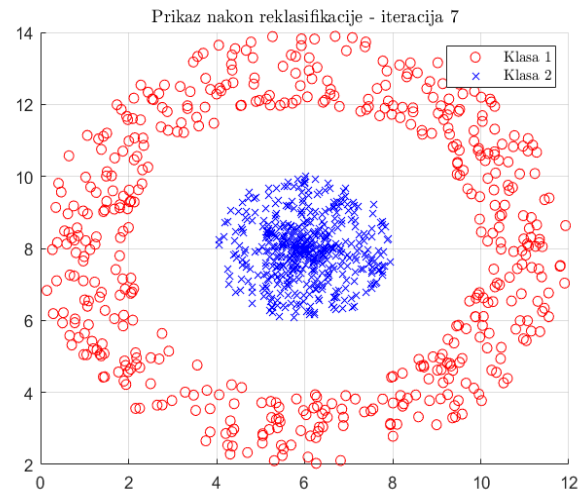
(a) Prikaz generisanih odbiraka u xy ravni



(b) Početna klasterizacija



(c) Neispravna reklasifikacija



(d) Ispravna reklasifikacija

Slika 19: Prikaz rada algoritma normalne dekompozicije

Na slici 19(a) prikazani su generisani odbirci dve klase koje koristimo u svrhu ovog zadatka.

Algoritam je veoma osetljiv na početnu klasterizaciju. Na slici 19(b) prikazana je početna klasterizacija koja je generisana na potpuno slučajan način. Kod takve početne klasterizacije pojavljuje se problem koji je prikazan na slici 19(c), a to je da se algoritam završi sa netačnom reklasterizacijom. Ovaj problem se može rešiti slično kao i kod prethodnih algoritama, a rešenje je da se u početnom trenutku veštački ubaci po 20% odbiraka u klastere kojima stvarno i pripadaju, a ostatak se preraspodeli na potpuno slučajan način. Rezultat takve klasterizacije prikazan je na slici 19(d). Možemo uočiti da sa takvom klasterizacijom klasteri mogu zameniti mesta, a takođe, smanjuje se vreme izvršavanja algoritma, odnosno, potreban je manji broj iteracija za reklasifikaciju. U proseku, srednja vrednost potrebnog broja iteracija za završetak klasterizacije u ova dva slučaja iznosi 9.28 kada početna klasterizacija nije potpuno slučajna, odnosno, 16.89 iteracija kada je potpuno slučajna, što je gotovo dvostruko duže, a pri tome konvergencija algoritma nije zagarantovana.

Algoritam je takođe jako osetljiv i na apriorno poznavanje broja klasa. U slučaju više klasa, jedan klaster će se podeliti na dva, stoga algoritam zahteva apriorno poznavanje broja klastera.